Leveraging Hyperbolic Embeddings for Coarse-to-Fine Robot Design

Heng Dong[1]*, Junyu Zhang[2]*, Chongjie Zhang[3]

[1]Tsinghua University, [2]Huazhong University of Science and Technology, [3]Washington University in St. Louis

Introduction

- Natural evolution changes the bodies of **real-world creatures** to allow them to solve daily tasks better.
- Can we mimic this evolution process so that **robots** can also solve tasks better by changing their morphologies?
  YES → **Robot Design**

Formulation of robot design and its challenges

- Robot design problem can be formulated as a bi-level optimization problem.

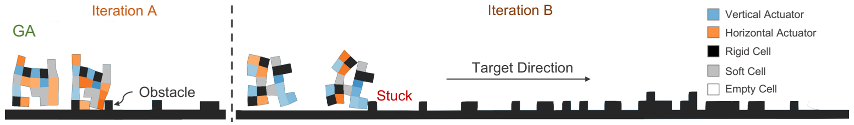
performance evaluation | control policy for design D | a robot design

$$D^* = \arg\max_D J(\pi_D, D)$$

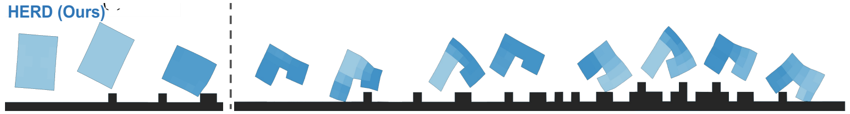
$$\text{subject to} \quad \pi_D = \arg\max_\pi J(\pi, D)$$

- Outer level: search in the **vast design space**
- Inner level: learn the optimal controller for each candidate design, which is **computationally expensive**

Previous work on robot design and our idea

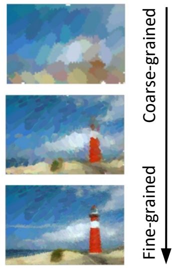


- Genetic Algo (GA) directly searches robots in the **vast design space** and **struggles to learn the controllers** for these complex robot designs. It fails to find an effective robot to cross the obstacles.



- Our method designs robots in a **coarse-to-fine** manner: first search for coarse-grained designs with satisfactory performance by ignoring smaller components and then iteratively refine them by adding/deleting/changing smaller components.
- For coarse-grained robots, **the design space is reduced and the controllers are easier to learn.**

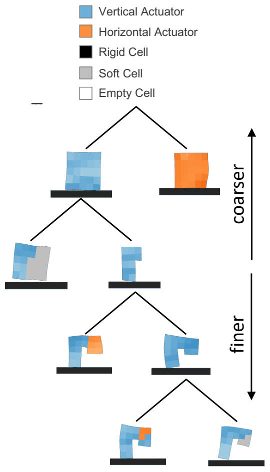
Intuition



- An intuitive example of the idea of coarse-to-fine from painting.
- Larger components first, then smaller components.

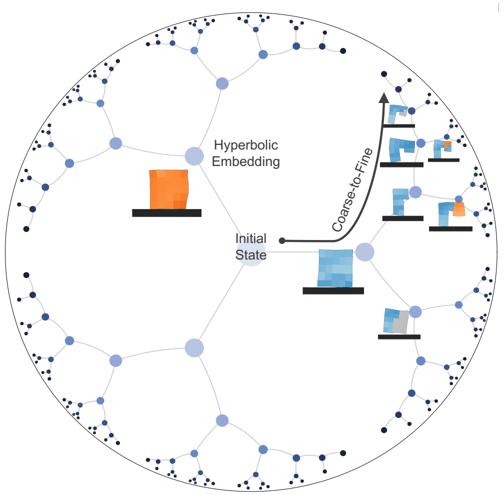
Method

1. Tree structure



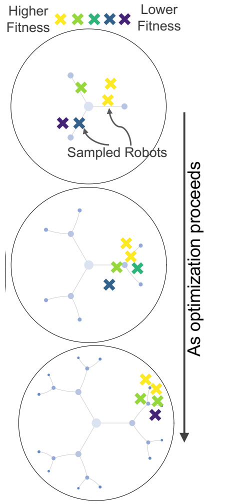
- The design space can be organized as a **tree**, where child robots are obtained by adding/deleting/changing smaller components of their parent robot.
- One simple method is to directly search for optimal robot in this tree. However, we have to determine whether to continue searching over coarse-grained robots or refine them, which could be troublesome.

2. Hyperbolic embeddings



- For better optimization property, we propose to embed the **discrete** tree of robots into a unified **continuous** space, i.e., hyperbolic space
- The area of a 2D Euclidean space grows **polynomially** w.r.t the radius.
- Hypernolic space: **exponentially.** It can be regarded as the continuous version of trees.

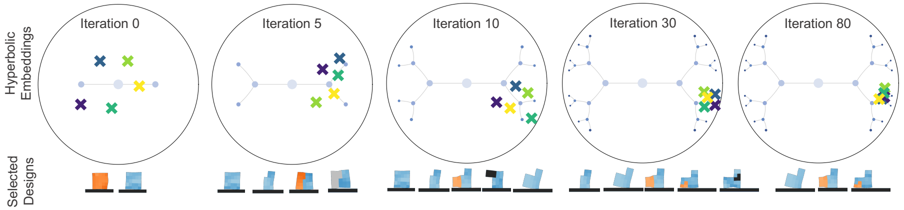
3. Optimization



- **Coarse-to-fine robot design** can then be simply implemented by automatically sampling robots from the center to the border of hyperbolic space.

Results

Visualization of the learning process



- As the sampled embeddings approach the border, the designed robots change from coarse-grained to fine-grained

Learning curve & replay